

**AMENDMENTS TO THE CLAIMS**

Claims 1-47 (Canceled)

48. (Currently Amended) A universal operating system stored in a memory of a computerized controller comprising a processor with said memory and the operating system stored in said memory, the controller further comprising a operating state storage, a nonvolatile storage, the computerized controller being operable to control at least one computerized wagering game, the operating system comprising:

an operating system kernel and a system handler application, the operating system kernel and system handler application operable to dynamically link with a plurality of program shared objects and device handlers and load said program shared objects and device handlers;

the system handler application further comprising an event queue;

the system handler application comprising an Application Program Interface (API) comprising functions callable from the program shared objects, the Application Program Interface comprising a plurality of operating functions callable by and used by at least some of the shared objects;

the system handler application operable to initiate a game based on data variables stored in the nonvolatile storage, the system handler application operable to write data variables to at least one of the operating state storage and nonvolatile storage; and

the device handlers accessing a look-up table in the operating state storage for the data variables stored in the nonvolatile storage, wherein changing the data variables in the nonvolatile storage causes execution of a corresponding callback function in one of the plurality of program shared objects of the system handler application.

49. (Previously presented) The universal operating system of claim 48, wherein the system handler application further comprises an event handler.

50. (Previously presented) The universal operating system of claim 48, wherein the system handler application comprises software having the ability when executed to:

unload a previous program shared object or device handler if a previous object or device handler has been loaded;

load a new program shared object or device handler; and

execute the new program shared object or device handler.

51. (Previously presented) The universal operating system of claim 48, wherein data variables modified by the program shared objects are stored by the system handler application in the nonvolatile storage and the operating state storage, and the system handler application functions to verify that the operating system or code for a shared object has not changed.

52. (Previously presented) The universal operating system of claim 51 wherein the operating state storage provides a variable name index to associated variable data locations within the nonvolatile storage.

53. (Canceled)

54. (Previously presented) The universal operating system of claim 48, further comprising a plurality of APIs.

55. (Previously presented) The universal operating system of claim 48, wherein the operating system kernel is a Linux operating system kernel.

56. (Previously presented) The universal operating system of claim 55, wherein the Linux operating system kernel has at least one selected device handler disabled.

57. (Previously presented) The universal operating system of claim 56, wherein the at least one selected device handler that is disabled is selected from the group consisting of a keyboard handler, an I/O port handler, a network interface handler, a storage device controller handler, and a I/O device handler.

58. (Previously presented) The universal operating system of claim 57, wherein the system handler application and the operating system kernel work in

communication to hash system handler application code and operating system kernel code.

59. (Previously presented) The universal operating system of claim 48 wherein the operating system is controlled by a general-purpose computer and the nonvolatile storage stores program variables, such that loss of power does not result in loss of the state of the computerized wagering game system, and the system handler application loads a first shared object and the first shared object calls up a function from within an Application Program Interface.

60. (Previously presented) The universal operating system of claim 59 wherein the system application handler loads and executes a single shared object at any one time but other shared objects are operable to share data via the program variables stored in nonvolatile storage.

61. (Previously presented) A method of managing data in a universal operating system via a system handler application, the method comprising:

(a) executing an operating system which then loads and operates an operating system kernel and a system handler application, both the operating system kernel and system handler application operable to dynamically link with a plurality of program shared objects and device handlers and load said shared objects and device handlers, the system handler application comprising an Application Program Interface comprising a plurality of functions callable from at least some of the shared objects, the system handler application operable to initiate a game based on data variables stored in a nonvolatile storage and the system handler application operable to write data variables to one of the nonvolatile storage or an operating state storage, the operating state storage comprising a look-up table for data variables stored in the nonvolatile storage,

(b) the system handler application then loading a first shared object and providing Application Program Interface functions called by the first shared object,

(c) the system handler application then executing the first shared object,

(d) the system handler application then storing data variables in the nonvolatile storage, such that a second shared object later or a first device handler

loaded can access the data variables in nonvolatile storage by utilizing the look-up table of the operating state storage,

- (e) the system handler application then unloading the first shared object,
- (f) the system handler application then loading other shared objects and repeating steps (b) through (e) for said other shared objects,
- (g) the operating system kernel then loading at least one additional shared object and repeating steps (b) through (e) for said at least one additional shared object,
- (h) the system handler application then loading at least one device handler and repeating steps (b) through (e) for said at least one device handler, and
- (i) the operating system kernel then loading at least one additional device handler and repeating steps (b) through (e) for said at least one additional device handler.

62. (Previously presented) The method of claim 61 further comprising the system handler application executing a corresponding callback function upon alteration of variable data in nonvolatile storage.

63. (Previously presented) The method of claim 62 further comprising handling events via the system handler application.